IN THE CLAIMS:

Kindly replace the claims of record with the following full set of claims:

1. (Currently amended) A battery powered device comprising:

a disk memory ~~(80)~~;

a means ~~(82)~~ for rotating the disk memory;

a read/write means ~~(84)~~ for at least one of reading and writing streaming data from or to the disk memory;

a buffer memory ~~(86)~~ for storing data read from or to be written to the disk memory, said buffer memory including a plurality of buffer memories associated with a corresponding one of said streaming data allowing concurrent reading and writing, wherein at least one buffer memory buffers data to be written and at least one buffer memory buffers data that has been read, said plurality of buffer memories being chosen proportional to a maximum bit rate associated with said corresponding streaming data,

an energy saving scheduling means ~~(96,100)~~ for:

monitoring the buffer memory to determine how long before each of said at least one memory buffer performing a write function is full and to determined how long before each of said at least one of memory buffer memory performing a read function is empty;

adjusting scheduled disk starting and stopping times in accordance with the monitored fullness of each of the plurality of the buffer memories, wherein said adjusting includes at least one of a time based reordering a refilling and emptying of each of said plurality of buffers to remove gaps between filling said memory buffers; and

controlling the disk memory rotating means in accordance with the adjusted schedule. ~~monitored buffer memory~~

2. (Cancelled)

3. (Cancelled)

4. (Currently amended) The device as set forth in claim [[3]] 1 further including:

a Back-Front-Back scheduling process ~~(100)~~ to reorder a refilling/~~emtying~~ emptying of various buffers and to remove gaps between buffer memory filling intervals.

5. (Cancelled)

6. (Currently amended) The device as set forth in claim 1 further including:

a system means ~~(90)~~ for controlling access to the memory means;

a means ~~(92)~~ for user input/output, in communication with the controller means ~~(90)~~;

a means ~~(94)~~ for coherently storing/reading the streams to/from the storage means ~~(80)~~, in communication with the controller means ~~(90)~~.

7. (Currently amended) The device as set forth in claim 1 wherein the buffer memory is partitioned into n buffers ~~(80)~~for n data streams.

8.(Currently amended) The device as set forth in claim 7 wherein the scheduling means performs the [[steps]] functions of:

spinning up a disk memory;

filling/emptying each stream i of a set S of the n data streams by reading/writing from/to the storage means until the respective stream is full/empty;

determining an earliest next spinning up time t_0;

putting the storage means in standby mode thereby spinning down the disk memory;

re-determining, at or just prior to time t_0, a new earliest spinning up time t_k based on current buffer fillings $f_i(t)$ at a current time t for each stream i in the set S;

iteratively performing the re-determining the new earliest spinning up time t_k until the time t_k is within a predetermined closeness to t_{k-1} or within a predetermined closeness to the current time t; and,

waiting until time t_k, or just prior to time t_k.

9. (Original) The device as set forth in claim 8, wherein the earliest next spinning up time t_0 is within at least $M / \sum_{j \in S} r_j^{\max}$ time units of the current time t, $r_i^{\max}$ being a maximum bit rate of the stream i, and m being a total amount of partitioned memory of the n buffers.

10. (Original) The device as set forth in claim 9, further including:

a means for examining the latest possible filling/emptying interval, $[s_i, e_i)$, for each stream i of the set S in order of non-increasing end time, s_i being a start time for the interval of stream i and e_i being an end time, including:

a means for advancing a preceding filling/emptying interval $[s_j, e_j)$ by an advance time $e_j - s_j$ if $e_j > s_j$; and,

a means for re-examining the latest possible filling/emptying interval, $[s_i, e_i)$, for each stream i of the set S in order of increasing end time, including:

a means for removing gaps between successive filling/emptying intervals by advancing intervals behind a gap forward in time, a gap existing whenever $s_{j+1} > e_j$.

11. (Currently amended) An energy efficient disk scheduling method comprising:

at least one of reading and writing streaming data from/to a disk memory;

buffering the data that is at least one of read from and written to the disk memory, wherein said buffering includes a plurality of buffer memories associated with a corresponding one of streaming data allowing concurrent reading and writing, wherein at least one buffer memory buffers data to be written and at least one buffer memory buffers data that has been read, said plurality of buffer memories being chosen proportional to a maximum bit rate associated with said corresponding streaming data;

monitoring the buffered data to determine how long before each of said at least one memory buffer performing a write function is full and to determined how long before each of said at least one of memory buffer memory performing a read function is empty:

adjusting scheduled disk starting and stopping times in accordance with the monitored fullness of each of the buffer memories, wherein said adjusting includes at least a time based reordering one of a refilling and emptying of each of said plurality of buffers to remove gaps between filling said memory buffers; and

scheduling rotation of the disk memory in accordance adjusted schedule ~~with the monitored buffered data~~.

12. (Cancelled)

13. (Cancelled)

14. (Currently amended) The method as set forth in claim 11, further including:

~~allocating a total amount M of memory for buffering stream data;~~

partitioning the memory M over a set S of n streams, each stream i of the set S being given a partitioned amount of buffer memory~~; and,~~

~~repeatedly performing scheduling~~.

15. (Original) The method as set forth in claim 14 wherein the scheduling includes:

spinning up a disk memory;

filling/emptying each stream i of set S by reading/writing from/to the storage means until the respective stream is full/empty;

determining an earliest next spinning up time t_0;

putting the storage means in standby mode thereby spinning down the storage means;

re-determining, at or just prior to time t_0, a new earliest spinning up time t_k based on current buffer fillings $f_i(t)$ at a current time t for each stream i in the set S;

iteratively performing the re-determining the new earliest spinning up time t_k until the time t_k is within a predetermined closeness to t_{k-1} or within a predetermined closeness to the current time t; and,

waiting until time t_k, or just prior to time t_k.

16. (Original) The method as set forth in claim 15, wherein the partitioned amount of memory for each stream i of the set S is approximately $r_i^{\max} M / \sum_{j \in S} r_j^{\max}$, $r_i^{\max}$ being a maximum bit rate of the stream i.

17. (Original) The method as set forth in claim 15, wherein the earliest next spinning up time t_0 is within at least $M / \sum_{j \in S} r_j^{\max}$ time units of the current time t, $r_i^{\max}$ being a maximum bit rate of the stream i.

18. (Original) The method as set forth in claim 15, wherein the total amount M of memory is solid-state memory.

19. (Original)The method as set forth in claim 15, further including controlling admission of streams to ensure that $\sum_{i \in S} r_i^{\max} < r_{\text{disk}}$, $r_i^{\max}$ being a maximum bit rate of the stream i, and r_{disk} being a rate of the storage means.

20. (Original)The method as set forth in claim 15, further including pausing between two successive streams, j and $j+1$ when all subsequent streams $j+1, j+2, ..., n$ can be delayed.

21. (Original) The method as set forth in claim 15, further including:

examining the latest possible filling/emptying interval, $[s_i, e_i)$, for each stream i of the set S in order of non-increasing end time, s_i being a start time for the interval of stream i and e_i being an end time, including:

advancing a preceding filling/emptying interval $[s_j, e_j)$ by an advance time $e_j - s_j$ if $e_j > s_j$; and,

re-examining the latest possible filling/emptying interval, $[s_i, e_i)$, for each stream i of the set S in order of increasing end time, including:

removing gaps between successive filling/emptying intervals by advancing intervals behind a gap forward in time, a gap existing whenever $s_{j+1} > e_j$.

22. (Original) The method as set forth in claim 21, further including repeatedly applying the examining and re-examining in each of the scheduling cycles.